# Grounded Question-Answering in Long Egocentric Videos

Shangzhe Di[1]     Weidi Xie[1,2]

[1]CMIC, Shanghai Jiao Tong University, China     [2]Shanghai AI Lab, China

## Abstract

*Existing approaches to video understanding, mainly designed for short videos from a third-person perspective, are limited in their applicability in certain fields, such as robotics. In this paper, we delve into open-ended question-answering (QA) in long, egocentric videos, which allows individuals or robots to inquire about their own past visual experiences. This task presents unique challenges, including the complexity of temporally grounding queries within extensive video content, the high resource demands for precise data annotation, and the inherent difficulty of evaluating open-ended answers due to their ambiguous nature. Our proposed approach tackles these challenges by (i) integrating query grounding and answering within a unified model to reduce error propagation; (ii) employing large language models for efficient and scalable data synthesis; and (iii) introducing a close-ended QA task for evaluation, to manage answer ambiguity. Extensive experiments demonstrate the effectiveness of our method, which also achieves state-of-the-art performance on the QAEgo4D and Ego4D-NLQ benchmarks. Code, data, and models are open-sourced [1].*

## 1. Introduction

In the literature, existing video perception tasks have primarily focused on videos in third-person view, for example, action recognition [4, 11, 14], video-language grounding [9, 16, 30], and video question-answering [17, 37, 40], these videos are short, *e.g.*, typically ranging from 10 seconds to one minute. Until recently, the proposal of Ego4D dataset [12] re-ignites the interest of video understanding from egocentric views, where the inputs are normally long, continuous video streams from the first-person point of view, *i.e.*, seeing the world through the eyes of an agent actively engaged with its environment, which resembles an important step towards deploying vision models into real-world scenarios, such as robotics and augmented reality.

In this paper, we consider question answering (QA) in long, egocentric videos, as illustrated in Fig. 1. Given ques-
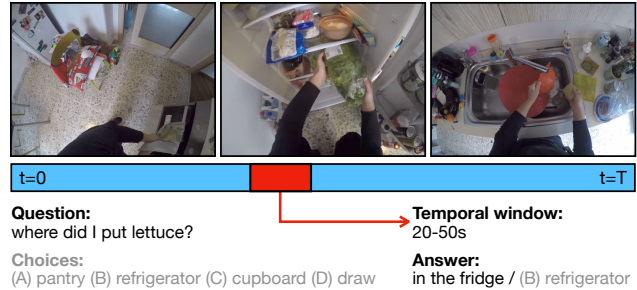


Figure 1. We propose a unified model for addressing grounded question answering in long egocentric videos, *i.e.*, simultaneously identifying the temporal window to a question, generating answers in natural language (OpenQA task), or picking answers from candidate choices (CloseQA task).

tions about an egocentric video, *e.g.*, "where did I put lettuce", we aim to build a visual system that can answer the raised question in free-form language. This task serves two purposes: enhancing episodic memory [35], *i.e.*, allowing a person or robot to ask questions on the fly about their own past visual experience; or probing the multi-modal reasoning abilities of deep models.

Question-answering (QA) in long egocentric videos is challenging, primarily due to the complexity of temporally grounding and generating answers to the queries within extensive video content. A pioneer work, overlooking the importance of query grounding, achieves unsatisfactory QA performance that merely outperforms "blind guessing" [3]. On the other hand, research about temporal grounding in long egocentric videos, while achieving good progress, is limited in practical uses without the QA ability. A potential fix would be chaining models from the two areas, *i.e.*, starting by localizing the temporal window to which the question relates, and followed by answering based on the corresponding video context. However, such a method is often ineffective due to error propagation. To address these challenges, we propose to train a unified model for simultaneous query grounding and answering, as shown in Fig. 1. The unified training has three advantages: *First*, by training the grounding task, the model can better grasp query-relevant information from the long videos, which is helpful for effective QA; *Second*, simultaneously training these two

---

[1]https://github.com/Becomebright/GroundVQA

tasks can reduce error accumulation thanks to the synergy effect [32] in deep models; *Third*, predicting a temporal window helps to understand the cause of failure. Thus, we propose to solve the query grounding and answering concurrently, namely **GroundVQA**.

Nevertheless, training the unified architecture demands significant resources and effort to manually annotate the triplets – comprising a question, answer, and temporal window – on lengthy videos. Limited training data poses a major challenge in training large models with millions of parameters. To combat this issue, we establish an automatic pipeline that leverages large language models (LLMs) to generate abundant training samples. This pipeline prompts LLMs to transform the plentiful, timestamped narrations in Ego4D into QA pairs, and estimates corresponding temporal windows. As a result, we produce 303K data samples from 5,389 video clips, which is a 30-fold increase over the existing dataset [3]. Our newly created pre-training dataset, named **EGOTIMEQA**, effectively mitigates overfitting and significantly enhances grounding and QA performance.

In addition, we face challenges in evaluating open-ended answers, *i.e.*, free-form language generation. Although open-ended QA is more representative of real-world scenarios where users interact with systems in natural language, it is a common consensus that the existing metrics like BLEU [27], METEOR [2], and ROUGE [18] are not fully satisfactory. To address this, we introduce CloseQA, an alternative close-ended task, where the model is asked to pick the correct answer from a set of candidate choices. We again leverage LLMs to generate plausible but incorrect answers, providing training and testing data for CloseQA.

The rest of the paper is structured as follows: Sec. 2 summarizes and discusses the relevant literature. Sec. 3 begins with an introduction to the proposed model for simultaneous query grounding and answering, followed by an automatic pipeline for augmenting the existing training dataset in a scalable manner. In Sec. 4, comprehensive ablation studies are presented to demonstrate the effectiveness of our proposed techniques. Consequently, our model achieves state-of-the-art performance on the QAEGO4D [3] and Ego4D-NLQ [12] benchmarks.

## 2. Related Work

**Video language grounding.** Video language grounding (VLG), initially proposed by Hendricks et al. [1], involves identifying and segmenting specific temporal intervals within third-person view videos based on a natural language description or query [20, 38, 43]. Several datasets and benchmarks, such as Charades-STA[9] and TACoS [30], have been curated to support research in this field. Notably, the Ego4D-NLQ [12] dataset features long-form egocentric videos paired with natural language queries. The NaQ dataset [29] further expands NLQ by re-

purposing the extensive narrations within Ego4D as queries, thereby enhancing model performance [13]. However, these narrations are not directly applicable to question-answering (QA) tasks. To bridge this gap, we introduce a generation pipeline that transforms these narrations into structured QA pairs. Additionally, we establish a multi-modal generative model capable of temporally localizing and answering a language query given long, egocentric videos.

**Video question answering.** Video question answering (VideoQA) entails generating responses to natural language queries by analyzing video content. This challenging task requires a detailed understanding of both visual and textual information. The advent of VideoQA datasets has catalyzed advancements in VideoQA research and benchmarking. For instance, ActivityNet-QA [40], which focuses on a variety of human activities, facilitates the evaluation of a model's ability to interpret complex actions and interactions. In contrast, How2QA [17], derived from instructional videos, emphasizes understanding sequential processes. NextQA [37] stands out by concentrating on causal and temporal reasoning in videos. These datasets typically include short videos, thereby limiting their relevance to real-world situations. In response, QAEGO4D [3] offers a long-form VideoQA benchmark featuring over a thousand egocentric videos with an average length of 8.2 minutes, each annotated with open-ended answers based on the aforementioned NLQ data. Our proposed method achieves state-of-the-art performance on this benchmark.

Annotating VideoQA datasets is labor-intensive and expensive [39], while insufficient training data often results in over-fitting. To address this, automatic generation of VideoQA data has been investigated. For example, JustAsk [39] generates QA pairs from transcribed speech using pre-trained language models, substantially expanding the dataset size. More recently, Large language models (LLMs) have shown remarkable proficiency in task processing and reasoning. Innovative studies like LLaVA [22] and MiniGPT-4 [46] leverage the powerful capabilities of LLMs to generate visual instruction tuning data, achieving notable success in a range of visual-language tasks. In our study, we exploit LLMs to transform existing narrations from the Ego4D dataset into question-answer pairs with temporal windows, facilitating multimodal understanding for egocentric videos. A concurrent work, EgoSchema [25], also exploits LLMs for constructing QA pairs. Compared to it, our approach includes both CloseQA and OpenQA, offering greater real-world applicability. Moreover, EgoSchema aims to summarize entire videos, while our method emphasizes episodic memory, focusing on recalling specific fragments for fine-grained queries.

**Egocentric video understanding.** Egocentric video understanding, a rapidly evolving field, focuses on analyzing videos captured by wearable cameras. This field
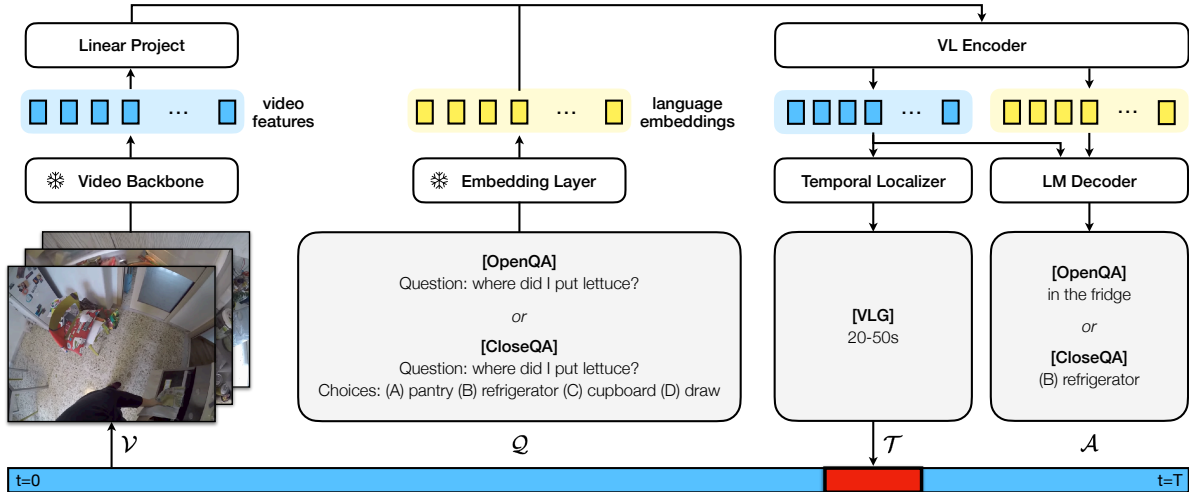
Figure 2. **Overview of GroundVQA**. It addresses three tasks: OpenQA, CloseQA, and VLG. The model processes a video $\mathcal{V}$ and a question $\mathcal{Q}$, to reason about the relevant temporal window $\mathcal{T}$ and the answer $\mathcal{A}$. Initially, a frozen video backbone encodes $\mathcal{V}$ and maps it into the language embedding space. Simultaneously, $\mathcal{Q}$ undergoes tokenization and is transformed through an embedding layer. These video and question embeddings are then fused using a visual-language encoder. Finally, a temporal localizer uses the resulting video features to predict $\mathcal{T}$, whereas a language decoder utilizes both video and question features, as provided by the VL encoder, to generate $\mathcal{A}$.

boosts a wide range of applications, including robotics, healthcare, augmented reality, and assistance for individuals with visual impairments. Various datasets are accessible to support research in this domain, including EPIC-KITCHENS [7], which contains videos of kitchen activities; Charades-Ego [33], featuring various everyday tasks; and Ego4D [12], which provides a global collection of diverse egocentric videos. These resources have raised emerging research problems such as human-object interaction [26], action recognition [15], and predictive modeling [10], *etc*. In this work, we delve into the complex task of grounded question answering, which demands temporally localizing a segment from an untrimmed egocentric video that corresponds to a given question, and producing an answer in natural language.

## 3. Method

This paper investigates the problem of grounded question answering in long egocentric videos, *i.e.*, the simultaneous localization and answering of questions. In Sec. 3.1, we begin by formally defining the task. In Sec. 3.2, we introduce our model, GroundVQA, that enables temporally grounding of visual questions and generates answers in either free-form language or a multi-choice format. In Sec. 3.3, we describe an automatic QA generation pipeline that leverages Large Language Models (LLMs) to transform narrations into QA pairs with temporal windows, a strategy proven to mitigate overfitting caused by limited training data in existing QA dataset on egocentric videos [3]. Lastly, in Sec. 3.4, we detail the multi-task training procedure for our model.

### 3.1. Task Definition

In general, we are interested in the task of generating open-ended answers to natural language questions, with an emphasis on the challenges of temporal grounding and contextual visual-language understanding.

Considering an egocentric video $\mathcal{V} \in \mathbb{R}^{N \times H \times W \times 3}$ and a question $\mathcal{Q} := \{q_1, q_2, \ldots, q_M\}$ as inputs – where $N$ denotes the number of frames, $H$ and $W$ are the dimensions of each frame, and $M$ is the length of the query – our objective is to construct a model $\Phi$ that simultaneously performs question grounding and answering:

$$[\mathcal{T}, \mathcal{A}] = \Phi(\mathcal{V}, \mathcal{Q}). \qquad (1)$$

The temporal window $\mathcal{T} := (s, e)$, defined by its start time $s$ and end time $e$, pinpoints a specific segment of the video that is most relevant to the posed question, aligning with the concept of Video Language Grounding (**VLG**). Moreover, $\mathcal{A}$ is the generated responses, which can be in free-form language for open-ended question answering (**OpenQA**) or selected from multiple choices for close-ended question answering (**CloseQA**). Our proposal involves the concurrent training of the model on these three tasks.

### 3.2. A Multi-tasking Architecture

In Fig. 2, we present the architecture of our proposed **GroundVQA**, comprising five main components: a language embedding layer, a video feature encoder, a linear projection layer, a visual-language encoder, and a dual-headed decoder for temporal localization and answer generation. This section describes each component in detail.

**Language embedding layer.** This layer transforms the to-

**(A) Prompt of generating QA pairs for OpenQA**

**System Prompt**
I will provide you with a series of narrations that depict my behavior. You should generate one QA pair based on the narrations in the format of {"Q": <question>, "A": <answer>} ...
*other detailed requirements*

**In-Context Examples**
C pours hot water from the frying pan in his left hand into the bowl in his right hand.
{"Q": "What did I pour in the bowl?", "A": "boiling water"}
... *other two examples*

**Current Input**
C searches through the cabinet. C closes the cabinet. C picks the tin from the cabinet. C places the tin on the counter.

{"Q":"Where was the tin before I took it?", "A": "at the cabinet"}

**(B) Prompt of generating wrong answers for CloseQA**

I'll provide a question and its correct answer. Generate three plausible, but incorrect, answers that closely resemble the correct one. Make it challenging to identify the right answer. No preamble, get right to the three wrong answers and present them in a list.

{"Q": "What did I pour in the bowl?", "A": "boiling water"}
["hot oil", "steamed milk", "warm broth"]
... *other two examples*

{"Q":"Where was the tin before I took it?", "A": "at the cabinet"}

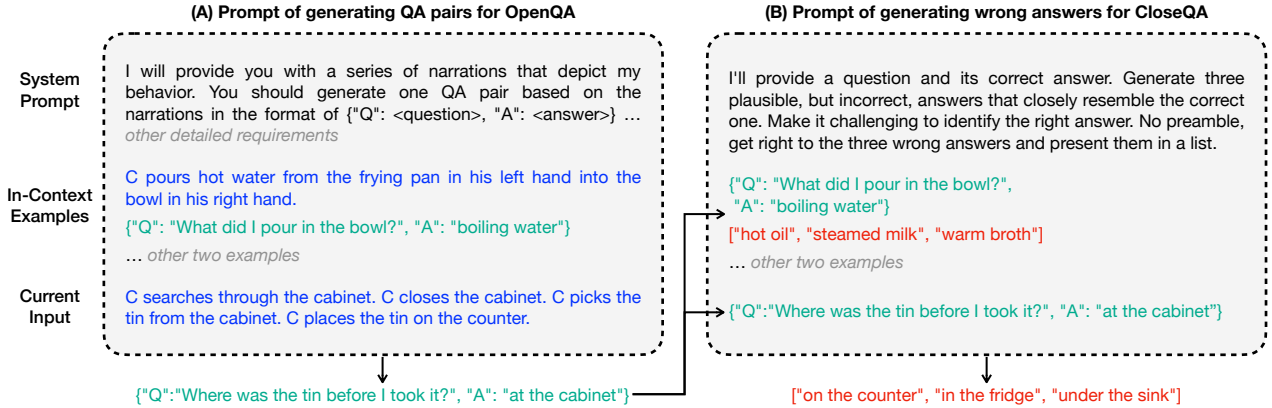["on the counter", "in the fridge", "under the sink"]

Figure 3. **The prompts for generating OpenQA and CloseQA training data with Llama2.** (A) First, we generate question-answer pairs using consecutive narration sentences from Ego4D. (B) Next, we generate three plausible yet incorrect answers for each question-answer pair to construct data for the CloseQA task. We provide in-context examples to enhance the generation quality.

kenized query into vector embeddings: $\mathcal{Q}' = \phi_{\mathrm{emb}}(\mathcal{Q})$. Specifically, in OpenQA, the term "query" refers to the questions being asked, whereas in CloseQA, a set of $K$ candidate answers is appended to the question.

**Video encoder and projection layer.** We utilize a frozen encoder, $\psi_{\mathrm{v}}$, to extract features from the video sequence. These features are then mapped to the language embedding space by a linear projection layer: $\mathcal{V}' = \phi_{\mathrm{proj}} \circ \psi_{\mathrm{v}}(\mathcal{V})$.

**Visual-language encoder.** Here, we use several Transformer encoder layers [36] that accept the projected video features and query embeddings as inputs, and fuse the visual-language information: $[\hat{\mathcal{Q}}, \hat{\mathcal{V}}] = \psi_{\mathrm{vl}}(\mathcal{Q}', \mathcal{V}')$.

**Temporal question localizer.** The objective here is to identify a temporal window within the video, that is most informative for answering the specific question. Our localizer takes the updated video feature from the visual-language encoder, and predicts the temporal window, *i.e.*, $\hat{\mathcal{T}} = \psi_{\mathrm{t}}(\hat{\mathcal{V}})$. Specifically, we adopt a similar module as in GroundNLQ [13] and ActionFormer [41], which consists of a classification head and a regression head. The classification head outputs a probability score for each timestamp's relevance to the question, while the regression head estimates the boundary distances from the current timestamp.

**Language decoder.** To generate answers to specific visual questions, we use a causal Transformer decoder. This decoder cross-attends to the output video and question features from the visual-language encoder and generates the answer in an auto-regressive manner: $\hat{\mathcal{A}} = \psi_{\mathrm{d}}(\hat{\mathcal{Q}}, \hat{\mathcal{V}})$.

### 3.3. Generate QA from Narrations

To train our model in concurrent query grounding and answering, as outlined in Equation 1, we utilize the Ego4D dataset [12]. This dataset comprises a vast collection of egocentric videos, each annotated with detailed, timestamped narrations describing the activities of the person wearing

the camera, with an average of 13.2 sentences per minute. Our goal is to exploit these high-quality narrations to create an automated pipeline that generates QA training samples using large language models (LLMs).

**Estimating temporal windows for narrations.** In an egocentric video $\mathcal{V}_i$, narrations are represented by the set $\{(\mathcal{N}_j, t_j)\}$, where $\mathcal{N}_j$ is a narration sentence and $t_j$ is its timestamp. To determine the temporal windows, we adopt a strategy akin to that in EgoVLP [19]:

$$\mathcal{T}_j = \left( t_j - \frac{\beta_i}{2\alpha}, \quad t_j + \frac{\beta_i}{2\alpha} \right), \tag{2}$$

where $\beta_i$ denotes the average interval between the timestamps of consecutive narrations, and $\alpha$ is the average of all $\beta_i$ values across videos. Essentially, these temporal windows are defined based on the dataset statistics.

**Generating OpenQA data.** We use an LLM to generate QA pairs from consecutive narration sentences. Considering that individual narration sentences are relatively short (7.4 words on average) and may lack sufficient information for generating meaningful questions, we propose to group consecutive sentences that collectively convey a complete context. Specifically, we segment the chronologically arranged narrations of a video into chunks. These chunks are based on either up to 5 sentences or a maximum duration of 30 seconds, whichever is reached first. For each chunk, we prompt the LLM to generate one QA pair and merge the associated temporal windows, resulting in a $(\mathcal{Q}, \mathcal{A}, \mathcal{T})$ pair. As depicted in Fig. 3 (A), the prompt comprises the chunk's narrations, detailed instructions, and three in-context examples to enhance the generation quality.

Utilizing the Llama2-13B model [34] on an NVIDIA A100 (80GB) GPU, we can generate approximately 20K QA pairs per hour, which is significantly more efficient than manual annotation. We apply this method to the training split of Ego4D$_{\mathrm{v2}}$ Episodic Memory dataset. Consequently,

we have created **EGOTIMEQA**, a grounded QA dataset containing 5,389 egocentric videos and 303K samples, as detailed in Tab. 1.

**Generating CloseQA data.** We prompt the LLM to generate three options that appear valid but are ultimately incorrect for a given question-answer pair. The constructed prompt is illustrated in Fig. 3 (B). We apply this procedure to augment EGOTIMEQA and QAEGO4D, enabling the training and evaluation of models in a multi-choice scenario. The generation speed reaches 40K samples per hour.

**Filtering CloseQA test set.** The LLM may generate implausible choices. To maintain the rigor of the CloseQA task, we filter out questions from the QAEGO4D test set that are easily answerable without video context. Specifically, we train a text-only "blind" model to identify and remove questions that are consistently answered correctly across ten trials with different seeds. Additionally, we perform rigorous human verification by eliminating samples that contain incorrect answers or temporal windows. The resulting **QAEGO4D$_{\mathbf{Close}}$** serves as a more refined testing ground. This ensures that models being evaluated truly require video content analysis to answer the questions correctly, thereby emphasizing the visual aspect of CloseQA.

### 3.4. Multi-task Training

Our model is designed to simultaneously address three tasks: open-ended question answering (OpenQA), close-ended question answering (CloseQA), and video-language grounding (VLG).

**Training for question-answering.** Training alternates between OpenQA and CloseQA to ensure proficiency in both question-answering formats. For OpenQA, inputs follow the format `question: <question>? video: <video feature>`. While for CloseQA, inputs are structured as `question: <question>? choices: <choices>. video: <video feature>`. To avoid memorization of answer positions, candidate answers are randomly shuffled. Moreover, the model is tasked to not only identify the correct option but also generate the associated answer, increasing training complexity. Cross-entropy loss is employed for both tasks, expressed as $\mathcal{L}_{QA} = \mathcal{L}_{ce}(\mathcal{A}, \hat{\mathcal{A}})$.

**Training for video-language grounding.** Concurrent with question-answering tasks, our model undergoes training on the VLG task. We employ temporal jittering [29] to augment temporal windows through random scaling and shifting. The loss function is a combination of binary Focal loss [21] and DIoU loss [45], formulated as $\mathcal{L}_{VLG} = \mathcal{L}_{focal}(\mathcal{T}, \hat{\mathcal{T}}) + \mathcal{L}_{DIoU}(\mathcal{T}, \hat{\mathcal{T}})$.

The final loss is a weighted sum: $\mathcal{L} = 0.5 \times \mathcal{L}_{VLG} + 0.5 \times \mathcal{L}_{QA}$. Incorporating the VLG task into training enhances the visual-language encoder's capability to distill relevant in-

| Dataset | # Video | # Sample | Supported Task | | |
|---|---|---|---|---|---|
| | | | OpenQA | CloseQA | VLG |
| train QAEGO4D | 997 | 11K | ✓ | ✓ | ✓ |
| train **EGOTIMEQA** | 5,389 | 303K | ✓ | ✓ | ✓ |
| val QAEGO4D | 162 | 1913 | ✓ | – | – |
| val NLQ$_{v2}$ | 415 | 4,552 | – | – | ✓ |
| test QAEGO4D | 166 | 1,850 | ✓ | – | – |
| test **QAEGO4D$_{Close}$** | 148 | 500 | – | ✓ | – |
| test NLQ$_{v2}$ | 333 | 4,004 | – | – | ✓ |

Table 1. **A summary of detailed dataset statistics.** Both the QAEGO4D and our EGOTIMEQA datasets support training on OpenQA, CloseQA, and VLG tasks. Hyper-parameters are picked based on the validating results on QAEGO4D and NLQ$_{v2}$, while models' performance is evaluated on corresponding test sets.

formation from videos, thereby boosting QA performance. Our model can be exclusively trained on VLG by freezing the LM decoder, or on a QA task by freezing the temporal localizer. Relevant experiments are provided in Sec. 4.4.

## 4. Experiments

### 4.1. Dataset and Metrics

**Natural Language Query (NLQ)** [12] is a prominent example of the video language grounding task. The second version of this benchmark, NLQ$_{v2}$, comprises 1,659 video clips, paired with 17.3K natural language queries and corresponding temporal windows. It is split into train, validation, and test sets, containing 11.3K, 3.9K, and 4K pairs, respectively. For evaluation, we use Recall@k, IoU=m, where $k \in \{1, 5\}$ and $m \in \{0.3, 0.5\}$. The primary metric for the NLQ challenge is **Mean Recall@1**, computed as the average of Recall@1, IoU=0.3 and Recall@1, IoU=0.5.

**NaQ** [29] augments NLQ by repurposing the extensive narrations with Ego4D as queries, including 5,389 video clips and 945K training samples.

**EGOTIMEQA** is our contributed pre-training dataset, containing the same video clips as NaQ, while featuring 303K question-answer pairs with temporal windows.

**QAEGO4D** [3] expands the NLQ benchmark by manually annotating open-ended answers on its train and validation sets. It consists of 1,325 video clips and 14.5K data samples, further divided into 10,746 training, 1,913 validation, and 1,850 testing samples. It adopts Accuracy and machine translation metrics including ROUGE-L (f-score) [18], METEOR [2], and BLEU-4 [27]. In our experiments, we exclude BLEU-4 because the majority (around $80\%$) of answers in QAEGO4D are under three words in length, and Accuracy as it's not effective due to language ambiguity in open-ended answers. To further address such ambiguity, we choose sentence similarity (**Sim.**) [31] as the primary metric, which maps sentences to a learned embedding space to

calculate cosine similarity. Specifically, we utilize the Sentence Transformers library and the `all-MiniLM-L6-v2` language model to perform the mapping.

**QAEGO4D_Close.** As detailed in Sec. 3.3, we have augmented QAEGO4D with a close-ended question answering (CloseQA) testing set. We run a model five times on this set with different seeds and calculate the **Accuracy** metric.

## 4.2. Implementation Details

**Video backbone features.** Recent studies, particularly InternVideo [5] and GroudNLQ [13], have utilized features from multiple video backbones to improve performance. To ensure a fair comparison, we use identical video features: EgoVLP, InternVideo-text, and InternVideo-verb. We concatenated these features along the channel dimension, forming 2304-dimensional feature vectors for each time step. Unless otherwise specified, we uniformly sample 1,200 vectors from these features as model input, which corresponds to an average of 8.2 minutes of video clips.

**Model configurations.** We use an instruction-tuned version of Flan-T5 [6, 28] as the language model. Our experiments involve its two variants: we conduct ablation studies using `Flan-T5-Small` (denoted as GroundVQA$_S$) and make final comparisons using `Flan-T5-Base` (denoted as GroundVQA$_B$). Our temporal localizer is adapted from ActionFormer [41], without using multi-scale pyramid features. This localizer comprises a classification head and a regression head, each has two layers of 1D convolution with layer normalization and ReLU activation in between.

**Training details.** We train all models with the AdamW optimizer [24], setting $\beta_1 = 0.9, \beta_2 = 0.999$, a learning rate of $1 \times 10^{-4}$, and no weight decay. The language embedding layer of Flan-T5 is fixed during training. Experiments are carried out on 4 NVIDIA A100 (80GB) GPUs, with gradient accumulation to maintain a consistent global batch size of 128. The training process is limited to 100 epochs, with early stopping based on the validation performance.

## 4.3. QA Baselines

As baseline models, we adopt the same models used in [3] and introduce several improvements for a fair comparison.

**BlindVQA** fine-tunes a `T5-Base` language model to answer questions without using video input. Essentially, BlindVQA serves as a language-only model to understand whether visual signals are essential to a specific question.

**SimpleVQA** enhances BlindVQA by incorporating visual capabilities. Here, video features are mapped to the language space and concatenated with question embeddings from an LM encoder. An LM decoder then generates answers given the merged features. Our proposed GroundVQA model differs from SimpleVQA, by conducting visual-language fusion in the encoder and adopting

VLG supervision on the fused video features.

**SimpleVQA+** builds on SimpleVQA by adding a ranking loss on LM Decoder's cross attention. Like our approach, SimpleVQA+ uses VLG supervision to emphasize the model's attention on question-relevant video segments. However, it cannot predict temporal windows, hindering the assessment of its grounding ability. Additionally, its performance falls short compared to our GroundVQA.

**Rehearsal Memory (RM)** [44] compresses long videos into a fixed-size memory. It segments a lengthy video into uniform parts, each processed by a Transformer encoder. Then, a recurrent module sequentially attends each segment feature to update the memory state. RM pretrains the memory state using reconstruction as a proxy task and further fine-tunes on the QA task.

**Improved baselines.** To ensure a fair comparison, we make several enhancements to the baseline models: (i) Replacing the original SlowFast [8] features with EgoVLP and InternVideo features, as specified in Sec. 4.2; (ii) Upgrading T5 to Flan-T5 and freezing word embeddings during training; (iii) Increasing the batch size to 128 and adjusting the learning rate to $1 \times 10^{-4}$. These modifications have consistently boosted the baseline model performance.

## 4.4. Ablations

In this section, we conduct experiments to investigate the effect of our proposal, for example, joint training of multiple tasks, integrating EGOTIMEQA, etc.

**Integrating the CloseQA task.** As presented in Tab. 2 (A-B), simultaneously training OpenQA and CloseQA tasks, despite their varying input-output formats, marginally impacts OpenQA performance. However, this integration offers a more comprehensive and reasonable method for assessing the model's question-answering capabilities. Thus, we integrate CloseQA in training as default.

**Integrating the VLG task.** As shown in Tab. 2, incorporating VLG task indeed improves question-answering performance, *e.g.*, GroundVQA$_S$'s Sentence Similarity increases from 54.8 to 55.6 when trained on QAEGO4D (B-C) and increases from 56.1 to 57.7 when trained on both QAEGO4D and EGOTIMEQA (D-E), demonstrating the effectiveness of our proposed multi-task training approach.

Conversely, SimpleVQA+$_S$ utilizes VLG supervision to direct the LM Decoder's cross attention towards question-related video segments. However, this approach results in diminished QA performance (G to I and H to J). This suggests that the complexity of the VLG task exceeds the capacity of the cross-attentions.

**Unified v.s. separate training.** An alternative to our unified model is training two separate models, one for temporal grounding and the other for question-answering on the

| | Model | Additional Data | Additional Task | | OpenQA | | | CloseQA |
|---|---|---|---|---|---|---|---|---|
| | | EGOTIMEQA | CloseQA | VLG | **Sim.** | ROUGE | METEOR | **Accuracy** |
| (A) | **GroundVQA$_\mathbf{S}$** | – | – | – | 54.9 | 27.9 | 18.8 | – |
| (B) | **GroundVQA$_\mathbf{S}$** | – | ✓ | – | 54.8 | 27.7 | 18.7 | 39.5±0.5 |
| (C) | **GroundVQA$_\mathbf{S}$** | – | ✓ | ✓ | 55.6 | 29.0 | 19.8 | 40.8±1.0 |
| (D) | **GroundVQA$_\mathbf{S}$** | ✓ | ✓ | – | 56.1 | 28.8 | 20.1 | 47.2±0.5 |
| (E) | **GroundVQA$_\mathbf{S}$** | ✓ | ✓ | ✓ | 57.7 | 30.2 | 21.2 | 48.7±0.4 |
| (F) | Oracle | ✓ | ✓ | – | 58.4 | 30.9 | 21.9 | 53.5±0.7 |
| (G) | SimpleVQA$_S$ | – | ✓ | – | 54.9 | 28.0 | 19.0 | 41.3±0.4 |
| (H) | SimpleVQA$_S$ | ✓ | ✓ | – | 56.1 | 28.8 | 20.2 | 47.1±0.3 |
| (I) | SimpleVQA+$_S$ | – | ✓ | ★ | 54.7 | 27.9 | 19.0 | 39.3±0.6 |
| (J) | SimpleVQA+$_S$ | ✓ | ✓ | ★ | 55.4 | 28.1 | 19.5 | 42.0±0.7 |

Table 2. **Ablation study on QAEGO4D and QAEGO4D$_\mathtt{Close}$ test sets.** "Additional Data": adding training data beyond QAEGO4D. "Additional Task": incorporating training tasks beyond OpenQA. "Sim.": the Sentence Similarity metric. "Oracle" represents a variant of GroundVQA$_S$, taking only question-relevant video segments as input to bypass the need for temporal grounding, thereby establishing the upper-bound performance. SimpleVQA+ leverages VLG supervision but cannot solve the VLG task, indicated by "★".

| Training | EGOTIMEQA | OpenQA | | | CloseQA |
|---|---|---|---|---|---|
| | | **Sim.** | ROUGE | METEOR | **Accuracy** |
| Two-stage | – | 54.7 | 27.3 | 18.4 | 39.3±0.8 |
| **Unified** | – | 55.6 | 29.0 | 19.8 | 40.8±1.0 |
| Two-stage | ✓ | 56.0 | 28.3 | 19.9 | 46.4±0.7 |
| **Unified** | ✓ | 57.7 | 30.2 | 21.2 | 48.7±0.4 |

Table 3. **Effect of the unified training method.** The "Two-stage" method separately trains two GroundVQA$_S$ models: one for the VLG task, and the other for QA tasks using relevant video segments. During inference, it uses the grounding results from the first model in the question-answering process of the second model.

| QAEGO4D | EGOTIMEQA | NLQ$_{v2}$+NaQ | **Mean R@1** | Mean R@5 |
|---|---|---|---|---|
| ✓ | – | – | 8.8 | 20.0 |
| ✓ | ✓ | – | 18.4 | 37.2 |
| ✓ | ✓ | ✓ | 20.9 | 42.5 |

Table 4. **Data scaling effect on the NLQ$_{v2}$ val set.** We train our GroundVQA$_S$ model on OpenQA, CloseQA, and VLG tasks with different training data, and evaluate its VLG performance.

grounded video clip. Results in Tab. 3 validate the effectiveness of our unified training method.

**Incorporating EGOTIMEQA data.** Our data generation method produces 303K samples, a 30-fold increase over the QAEGO4D training set, resulting in notable performance gains in QA and VLG tasks. The QA metrics for GroundVQA$_S$ demonstrate significant enhancements, as evidenced in Tab. 2 (B to D and C to E). Similar improvements are observed for SimpleVQA$_S$ (G-H) and SimpleVQA+$_S$ (I-J), confirming the generality and effectiveness for EGO-TIMEQA. In Tab. 4, EGOTIMEQA also boosts VLG recall by a large margin, which is further amplified with NLQ and NaQ data. Notably, as depicted in Fig. 4, the value of EGO-TIMEQA is even more evident in overcoming overfitting.

Combining the above enhancements (A-E in Tab. 2), our method closely approaches the oracle upper bound (F), with
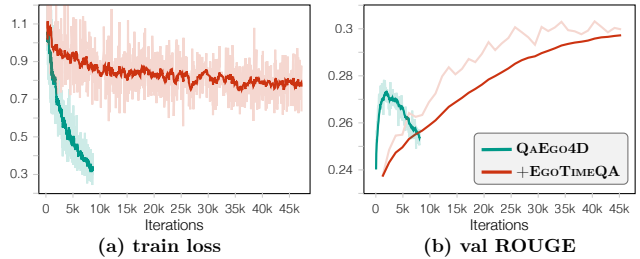


(a) train loss     (b) val ROUGE

Figure 4. **Training and validation curves of GroundVQA$_\mathbf{S}$.** The limited training data of QAEGO4D results in severe overfitting, which is effectively mitigated by our generated EGOTIMEQA.

the main gap due to imperfect temporal grounding.

## 4.5. Comparison with State-of-the-art

In this section, we compare our model to the state-of-the-art on OpenQA, CloseQA, and VLG tasks, and present qualitative examples in Fig. 5.

**On QAEGO4D.** We report results on the QAEGO4D test set in Tab. 5. To ensure fairness, we reproduce the other methods using identical settings (detailed in Sec. 4.3). Our model achieves the best performance, outperforming prior works by a large margin.

**On NLQ$_{v2}$.** We then assess VLG performance on the NLQ$_{v2}$ test set. As seen in Tab. 6, our model, without complex design or multi-scale feature pyramids [13], matches the SOTA performance. GroundVQA$_B^{\dagger}$ exhibits further improvements by pre-training on NLQ$_{v2}$ and NaQ, and fine-tuning exclusively on NLQ$_{v2}$.

**Qualitative analysis.** Fig. 5(A) shows an OpenQA example. Our model successfully predicts the temporal window and the answer, while SimpleVQA* fails. Although our predicted answer slightly differs from the ground truth, it's still valid, highlighting the challenge of paraphrasing in evaluating open-ended answers, thus reflecting the advantage of our CloseQA task. Fig. 5(B) demonstrates a CloseQA ex-
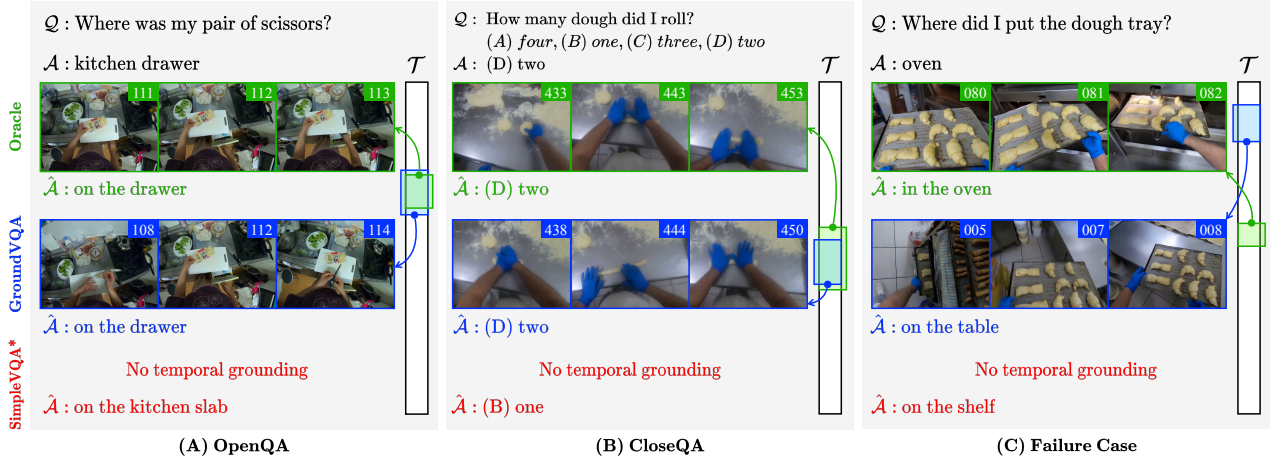
**Figure 5. Qualitative examples.** In our demonstration, we compare three models: the Oracle baseline, our GroundVQA, and SimpleVQA*. Each column presents a sample that includes the query $\mathcal{Q}$, the ground truth answer $\mathcal{A}$, three frames from the grounded video segment, and the predicted answer $\hat{\mathcal{A}}$. Additionally, each column illustrates the video's time span and the predicted temporal window $\mathcal{T}$, with Oracle's temporal window serving as the ground truth. Note that SimpleVQA* is incapable of predicting the temporal window.

| Method | OpenQA | | | CloseQA | Param |
|---|---|---|---|---|---|
| | Sim. | ROUGE | METEOR | Accuracy | |
| BlindVQA | - | 25.9 | 17.4 | - | 247 |
| BlindVQA* | 53.8 | 27.5 | 18.4 | 36.3±0.5 | 247 |
| SimpleVQA | - | 26.1 | 17.4 | - | 249 |
| SimpleVQA* | 55.7 | 28.6 | 19.3 | 41.1±0.5 | 249 |
| SimpleVQA+ | - | 27.1 | 18.3 | - | 249 |
| SimpleVQA+* | 55.7 | 28.8 | 19.5 | 41.4±0.3 | 249 |
| RM | - | 26.6 | 17.7 | - | 368 |
| RM* | 54.1 | 27.3 | 18.5 | 39.9±0.8 | 368 |
| **GroundVQA$_\mathrm{B}$** | **58.2** | **30.4** | **21.5** | **50.2±0.5** | 252 |

Table 5. **Comparison with the state of the art on QAEGO4D and QAEGO4D$_{\mathtt{Close}}$ test sets.** "Param": number of parameters in millions. Gray results are reported in [3] while "*" denotes our reproducing performance with several enhancements. "BlindVQA" represents the lower-bound baseline, learning only language bias.

| Method | Recall@1 | | | Recall@5 | |
|---|---|---|---|---|---|
| | Mean | IoU=0.3 | IoU=0.5 | IoU=0.3 | IoU=0.5 |
| VSLNet [42] | 4.08 | 5.42 | 2.75 | 8.79 | 5.07 |
| EgoVLP [19] | 8.35 | 10.46 | 6.24 | 16.76 | 11.29 |
| ReLER [23] | 10.51 | 12.89 | 8.14 | 15.41 | 9.94 |
| NaQ++ [29] | 17.67 | 21.70 | 13.64 | 25.12 | 16.33 |
| GroundNLQ [13] | 20.91 | 24.50 | 17.31 | **40.46** | **29.17** |
| **GroundVQA$_\mathrm{B}$** | 19.31 | 23.65 | 14.96 | 36.19 | 24.58 |
| **GroundVQA$_\mathrm{B}^{\dagger}$** | **22.15** | **26.67** | **17.63** | 39.94 | 27.70 |

Table 6. **Comparison with the state of the art on the NLQ$_{\mathrm{v2}}$ test set.** "GroundVQA$_\mathrm{B}$" is simultaneously trained on all three tasks with QAEGO4D and EGOTIMEQA data. "GroundVQA$_\mathrm{B}^{\dagger}$" follows NaQ++ and GroundNLQ, pre-trained solely on the VLG task with NLQ$_{\mathrm{v2}}$ and NaQ data, and further fine-tuned on NLQ$_{\mathrm{v2}}$.

ample. Our model shows competence in predicting a close temporal window and identifying the correct answer. On the contrary, SimpleVQA* chooses an incorrect answer, while the absence of temporal localization hinders understanding of its error source. Fig. 5(C) is a failure case of our model and SimpleVQA*. Yet, our model's temporal window prediction is relevant to the query, and the predicted answer is coherent with the grounded content. This highlights an issue of the QAEGO4D and NLQ annotations, where multiple relevant video segments and plausible answers exist, but only one annotation is available per query.

We present additional results in the supplementary material, including the impact of using different LLMs to generate QA data, a more in-depth statistical analysis of EGO-TIMEQA, additional qualitative findings, prompts for generating QA data, limitations, and future work.

## 5. Conclusion

In conclusion, this paper tackles the challenge of grounded question answering in long egocentric videos. We demonstrate the crucial role of precise temporal grounding in effective question-answering and propose a novel, unified model that concurrently tackles both tasks. To counter the risk of overfitting due to limited training data, we introduce an automated pipeline for generating extensive question-answer pairs from narrations using LLMs. Additionally, to address the challenge of evaluating open-ended answers, we present the CloseQA benchmark, ensuring more reliable evaluations. Extensive ablation studies confirm the effectiveness of our approach, which achieves state-of-the-art performance on the QAEGO4D and the Ego4D-NLQ benchmarks, marking a significant advancement in the field of egocentric video understanding.

# References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. 2, 5

[3] Leonard Bärmann and Alex Waibel. Where did i leave my keys? - episodic-memory-based question answering on egocentric videos. In *CVPR Workshop*, 2022. 1, 2, 3, 5, 6, 8

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1

[5] Guo Chen, Sen Xing, Zhe Chen, Yi Wang, Kunchang Li, Yizhuo Li, Yi Liu, Jiahao Wang, Yin-Dong Zheng, Bingkun Huang, et al. Internvideo-ego4d: A pack of champion solutions to ego4d challenges. In *ECCV Workshop*, 2022. 6

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv:2210.11416*, 2022. 6

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 3

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 6

[9] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 1, 2

[10] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *ICCV*, 2021. 3

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1

[12] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1, 2, 3, 4, 5

[13] Zhijian Hou, Lei Ji, Difei Gao, Wanjun Zhong, Kun Yan, Chao Li, Wing-Kwong Chan, Chong-Wah Ngo, Nan Duan, and Mike Zheng Shou. Groundnlq@ ego4d natural language queries challenge 2023. In *CVPR Workshop*, 2023. 2, 4, 6, 7, 8

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 1

[15] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 3

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 1

[17] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *EMNLP*, 2020. 1, 2

[18] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 2, 5

[19] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. In *NeurIPS*, 2022. 4, 8

[20] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *ICCV*, 2023. 2

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2

[23] Naiyuan Liu, Xiaohan Wang, Xiaobo Li, Yi Yang, and Yueting Zhuang. Reler@ zju-alibaba submission to the ego4d natural language queries challenge 2022. In *CVPR Workshop*, 2022. 8

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017. 6

[25] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *arXiv:2308.09126*, 2023. 2

[26] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *ICCV*, 2019. 3

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 2, 5

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020. 6

[29] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Naq: Leveraging narrations as queries to supervise episodic memory. In *CVPR*, 2023. 2, 5, 8

[30] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. In *ACL*, 2013. 1, 2

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019. 5

[32] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv:1706.05098*, 2017. 2

[33] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 3

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023. 4

[35] Endel Tulving et al. Episodic and semantic memory. *Organization of memory*, 1972. 1

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[37] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 1, 2

[38] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *ICCV*, 2023. 2

[39] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 2

[40] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 1, 2

[41] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, 2022. 4, 6

[42] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *ACL*, 2020. 8

[43] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *NeurIPS*, 2022. 2

[44] Zhu Zhang, Chang Zhou, Jianxin Ma, Zhijie Lin, Jingren Zhou, Hongxia Yang, and Zhou Zhao. Learning to rehearse in long sequence memorization. In *ICML*, 2021. 6

[45] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI*, 2020. 5

[46] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023. 2